Subject: Re: HDF, netCDF, etc question
Posted by Randall Skelton on Sat, 07 Jul 2001 13:07:55 GMT
View Forum Message <> Reply to Message

Subject: Re: HDF, netCDF, etc question
Posted by Craig Markwardt on Sun, 08 Jul 2001 03:40:21 GMT
View Forum Message <> Reply to Message

ronn kling <ronn@rlkling.com> writes:

> Hi All,
>
> Which format would be best for a large number of large sequential images
> along with ancillary data such as field of view, exposure time, etc?
> Discrimators would be things like speed in reading them in, ease to pull out
> the images and information.
>
> I don't have a lot of experience with these things so any and all opinions
> are welcomed.

Hi Ronn--

A similar question to this was asked a few months ago. No really deep technical discussions ensued, but this was the gist of it:

* I advocated astronomy's FITS format. Plus: platform independent, metadata is in ASCII, good support in IDL Astronomy Library. Minus: seen as "archane."

* I also advocated IDL SAVE files. With my library you can read and write SAVE files sequentially like any other file. Plus: native to IDL. Minus: tied to IDL.

* Many people seem to swear by Liam Gumley's binary tools. Plus: raw speed and direct access; platform neutral. Minus: low level.

* Martin Schultz advocated GRIB, half tongue in cheek.

As I said, no real answers came out of this, but at least it may give you some other ideas.

Craig

--
---------- ----------

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---

Subject: Re: HDF, netCDF, etc question
Posted by R.Bauer on Sun, 08 Jul 2001 16:31:58 GMT
View Forum Message <> Reply to Message

ronn kling wrote:
>
> Hi All,
>
> Which format would be best for a large number of large sequential images
> along with ancillary data such as field of view, exposure time, etc?
> Discrimators would be things like speed in reading them in, ease to pull out
> the images and information.
>
> I don't have a lot of experience with these things so any and all opinions
> are welcomed.
>
> Thanks,
> Ronn
>

Dear Ronn,

we have a lot of experience by netCDF and HDF (SD). Both you can use up to 8 dimensions
for data and you can store several ancillary data like attributes or parameters.

We have routines reading whole datasets or subsets which are able to concatinate
files during reading by a dimension name. (The next version of our library
will be published soon)

An example of our data structure is shown below. The tags what's in the structure
are defined and what's written to a file is given from a definition file. In the
example it's def_icg. You can define your own definitions and sumbit your definitions
by a keyword to each routine.

Most of our files have something between 2 MB and 200MB. A small amount

of files exceeds
700 MB. At the moment I am using only the SD interface from HDF. So I
have no experience
for the image interface. But this should not be very complicated.

best regards
Reimar

--

This template was created by write_icgspro.

```
PRO ronn

  ; ******************************************
  ; CALLED BY:
  ;
 write_icgspro,'ronn.pro',/small,/status,short=['time','LAT', 'LON','O3']
  ; ******************************************

  extra= { $
    SHORT:['time',$
           'LAT',$
           'LON',$
           'O3'] ,$
    SMALL:       1 ,$
    STATUS:      1 $
       }

  struct=gen_icgs(_extra=extra)

  ;   ******************************************
  ;      FILL IN YOUR DATA
  ;   ******************************************


 *STRUCT.!GLOBAL.!DEF.FILE='/usr/local/icg/icg/idl_source/idl _links/def_icg.pro'
  *STRUCT.!GLOBAL.!DEF.TIME='2001-06-29 06:00:30'
  *STRUCT.!GLOBAL.PI.NAME='R.Kling'
  *STRUCT.!GLOBAL.PI.ORGANISATION='KRS, inc.'
  *STRUCT.!GLOBAL.DATASET.TITLE='Test data interface'
  *STRUCT.!GLOBAL.DATASET.DATA_CATEGORY='EXPERIMENT'
  *STRUCT.!GLOBAL.DATASET.EXPERIMENT='RK-Test1'
  *STRUCT.!GLOBAL.DATASET.TYPE_OF_DATA='SIMULATION'
  *STRUCT.!GLOBAL.PLATFORM.TYPE='COMPUTER'
  *STRUCT.TIME.UNITS='seconds since 2000-01-01 00:00:00 UTC'
```

```
   *STRUCT.TIME.LONG_NAME='time'
   *STRUCT.TIME.FILL_VALUE='UNDEFINED'
   *STRUCT.TIME.FLAG='NONE'
   *STRUCT.TIME.PARAM=dindgen(10)
   *STRUCT.TIME.STATUS.DIM.NAME='time'
   *STRUCT.TIME.SHORT_NAME='time'

   *STRUCT.LAT.UNITS='deg'
   *STRUCT.LAT.LONG_NAME='longitude'
   *STRUCT.LAT.FILL_VALUE= -999
   *STRUCT.LAT.FLAG='NONE'
   *STRUCT.LAT.PARAM=findgen(4)
   *STRUCT.LAT.STATUS.DIM.NAME='LAT'
   *STRUCT.LAT.SHORT_NAME='LAT'

   *STRUCT.LON.UNITS='deg'
   *STRUCT.LON.LONG_NAME='LONGITUDE'
   *STRUCT.LON.FILL_VALUE= -999
   *STRUCT.LON.FLAG='NONE'
   *STRUCT.LON.PARAM=findgen(5)
   *STRUCT.LON.STATUS.DIM.NAME='LON'
   *STRUCT.LON.SHORT_NAME='LON'

   *STRUCT.O3.UNITS='ppm'
   *STRUCT.O3.LONG_NAME='mixing ratio'
   *STRUCT.O3.FILL_VALUE= -999
   *STRUCT.O3.FLAG='NONE'
   *STRUCT.O3.PARAM=findgen(4,5,10)
   *STRUCT.O3.STATUS.DIM.NAME=['LAT','LON','time']
   *STRUCT.O3.SHORT_NAME='O3'


   ; ****************************************
   ; FILL IN YOUR COMMANDS
   ; e.g.
   ; struct=chk_struct(struct)  ; to remove UNDEFINED attributes
   ;
   ; Write a netCDF-File:
   ; write_ncdf,'file.nc',struct ; /ignore_test ; /remove_previous ;
/overwrite
   ;
   ;
   ; Write a HDF-File:
   ; write_hdf,'file.hdf',struct ; /ignore_test ; /remove_previous ;
/overwrite
   ;
   ; Write an ENZ-FILE:
   ; struct=ptr_struct2struct(struct,/free) ; to convert a ptr_struct
```

```
into a normal structure
   ; write_enz,'file.enz',struct
   ; ****************************************
   ;

   free_ptr_struct,struct
END
```

> --
> Ronn Kling
> KRS, inc.
> email: ronn@rlkling.com
> "Application Development with IDL" programming book updated for IDL5.4!
> "Calling C from IDL, Using DLM's to extend your IDL code" NEW BOOK!
> Shareware and Freeware at: http://www.rlkling.com/

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html

Subject: Re: HDF, netCDF, etc question
Posted by Aaron Birenboim on Mon, 16 Jul 2001 20:38:31 GMT
View Forum Message <> Reply to Message

Randall Skelton <rhskelto@atm.ox.ac.uk> wrote:
: I suggest the use of an SQL database :-)

Frankly, I agree.

I have used many formats, including a customer that
demanded a customized HDF based representation.

All a big mess.

I highly recommend storing all your meta-data in an SQL database.
If you do not want to use BLOB's, store path names to the raw binary data.
Store data type, size, etc in the DB.

HDF is very flexible. So flexible that its not very useful
as a standard. In fact, we implemented on HDF 4, then HDF 5 came out,
and its not backward compatible. So this major customer
is sitting on top of TONS of data stored in an OBSOLETE format.
--
Aaron Birenboim
Albuquerque, NM
aaron@boim.com
http://boim.com/~aaron

Subject: Re: HDF, netCDF, etc question
Posted by Aaron Birenboim on Mon, 16 Jul 2001 20:40:27 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote:

: ronn kling <ronn@rlkling.com> writes:

: * I advocated astronomy's FITS format. Plus: platform independent,
: metadata is in ASCII, good support in IDL Astronomy Library. Minus:
: seen as "archane."

I agree. Its very simple, very "sharable".

: * Many people seem to swear by Liam Gumley's binary tools. Plus: raw
: speed and direct access; platform neutral. Minus: low level.

I dunno about this. But before I advocated a simple, binary format
tied to a database for metadata. These could be a help here.

--
Aaron Birenboim
Albuquerque, NM
aaron@boim.com
http://boim.com/~aaron

Subject: Re: HDF, netCDF, etc question
Posted by R.Bauer on Thu, 19 Jul 2001 06:20:50 GMT
View Forum Message <> Reply to Message

Aaron Birenboim wrote:
>
> Randall Skelton <rhskelto@atm.ox.ac.uk> wrote:
> : I suggest the use of an SQL database :-)
>
> Frankly, I agree.
>
> I have used many formats, including a customer that
> demanded a customized HDF based representation.
>
> All a big mess.
>
> I highly recommend storing all your meta-data in an SQL database.
> If you do not want to use BLOB's, store path names to the raw binary data.
> Store data type, size, etc in the DB.
>
> HDF is very flexible. So flexible that its not very useful
> as a standard.

I agree HDF or netCDF is very flexible. And if you don't give rules
to the people using it you get very different files for the same data
by each people.

How to give rules?

Our routines are using definition files e.g.

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ work/rb_lib/def_icg.pro
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ work/rb_lib/def_nilu.pro

Only attributes which are defined will be read or written. (case
sensitive)
If they are global or parameter attributes is defined
and the resulting structure is defined by this rules too.

David Stern told me last year that they don't upgrade
HDF till the problem of the versions by HDF is fixed.
So idl user should not get into this problems.

> In fact, we implemented on HDF 4, then HDF 5 came out,
> and its not backward compatible. So this major customer
> is sitting on top of TONS of data stored in an OBSOLETE format.
> --

> Aaron Birenboim
> Albuquerque, NM
> aaron@boim.com
> http://boim.com/~aaron

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html